

(19)



Europäisches  
Patentamt  
European  
Patent Office  
Office européen  
des brevets



(11)

EP 1 256 007 B1

(12)

## EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention  
of the grant of the patent:  
**25.06.2008 Bulletin 2008/26**

(51) Int Cl.:  
**G01R 31/28 (2006.01)**

(21) Application number: **00979203.7**

(86) International application number:  
**PCT/US2000/031780**

(22) Date of filing: **16.11.2000**

(87) International publication number:  
**WO 2001/038890 (31.05.2001 Gazette 2001/22)**

**(54) DECOMPRESSOR/PRPG FOR APPLYING PSEUDO-RANDOM AND DETERMINISTIC TEST PATTERNS**

DEKOMPRESSIOMSGERÄT/ PRPG ZUR ERZEUGUNG VON PSEUDO-ZUFÄLLIGEN UND DETERMINISTISCHEN PRÜFMUSTERN

DECOMPRESSEUR/PRPG POUR APPLICATION DE SEQUENCES DE TEST PSEUDO-ALEATOIRES ET DETERMINISTIQUES

(84) Designated Contracting States:  
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR**

(56) References cited:  
**US-A- 5 694 402 US-A- 5 991 898  
US-A- 5 991 909**

(30) Priority: **23.11.1999 US 167137 P  
15.11.2000 US 713664**

- RAJSKI J ET AL: "TEST DATA DECOMPRESSION FOR MULTIPLE SCAN DESIGNS WITH BOUNDARY SCAN" IEEE TRANSACTIONS ON COMPUTERS, IEEE INC. NEW YORK, US, vol. 47, no. 11, November 1998 (1998-11), pages 1188-1200, XP000831123 ISSN: 0018-9340
- ZACHARIA N ET AL: "Decompression of test data using variable-length seed LFSRs" 7 April 1992 (1992-04-07), PROCEEDINGS 13TH IEEE VLSI TEST SYMPOSIUM (CAT. NO.95TH8068) IEEE COMPUT. SOC. PRESS LOS ALAMITOS, CA, USA, PAGE(S) 426 - 433 , XP002303739 ISBN: 0-8186-7000-2 \* abstract; figure 6 \* section 5 "Hardware Implementation"
- LEW YAN VOON L F C ET AL: "BIST linear generator based on complemented outputs" 7 April 1992 (1992-04-07), VLSI TEST SYMPOSIUM, 1992. '10TH ANNIVERSARY. DESIGN, TEST AND APPLICATION: ASICS AND SYSTEMS-ON-A-CHIP', DIGEST OF PAPERS., 1992 IEEE ATLANTIC CITY, NJ, USA 7-9 APRIL 1992, NEW YORK, NY, USA, IEEE, US, PAGE(S) 137-142 , XP010063459 ISBN: 0-7803-0623-6 \* abstract; figure 1 \* section 1 "Introduction" section 2 "Mixed-Output ALSFM Theory" section 5 "BIST Application"

(43) Date of publication of application:  
**13.11.2002 Bulletin 2002/46**

(73) Proprietor: **Mentor Graphics Corporation  
Wilsonville, OR 97070-7777 (US)**

(72) Inventors:
 

- RAJSKI, Janusz  
West Linn, OR 97068 (US)
- TYSZER, Jerzy  
PL-61-249 Poznan (PL)
- KASSAB, Mark  
Wilsonville, OR 97070 (US)
- MUKHERJEE, Nilanjan  
Wilsonville, OR 97070 (US)

(74) Representative: **Hoarton, Lloyd Douglas Charles et al  
Forrester & Boehmert  
Pettenkoferstrasse 20-22  
80336 München (DE)**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

**Description****TECHNICAL FIELD**

5 [0001] This invention relates generally to testing of integrated circuits and, more particularly, to the generation and application of test data in the form of patterns, or vectors, to scan chains within a circuit-under-test.

**BACKGROUND**

10 [0002] Built-in self-test (BIST) is emerging as an attractive alternative to conventional methods of testing microelectronics devices at time of manufacture. In BIST, additional circuitry is added to a circuit-under-test to provide on-chip test-pattern generation, test-response evaluation, and test control. Consequently, BIST is significantly changing the way that integrated circuits are tested. It reduces testing cost by shortening test application time, minimizing the amount of test data that must be stored by an external tester, and reducing tester costs. Its implementation can result in a reduction  
15 of the product development cycle and a reduction in the cost of system maintenance.

[0003] The basic BIST objectives are on-chip test-pattern generation and test-response compaction. By far the most commonly-used means for generating test patterns on chip are pseudo-random test pattern generators (PRPGs). A PRPG generates a set of test patterns based on an initial value, or seed, loaded into memory elements within the PRPG. The popularity of pseudo-random tests stems from the very simple hardware required to implement the test generation.

20 The two principal forms of PRPGs, which evolved over time and are now commonly in use, are both linear finite state machines: linear feedback shift registers (LFSRs) and one-dimensional linear hybrid cellular automata (LHCAs).

[0004] Typically, an LFSR consists of interconnected memory elements (also referred to as flip-flops, stages, cells, etc.) and linear logic elements (such as XOR or XNOR gates). An LFSR of length  $n$  can be also represented by its characteristic polynomial  $h_nx^n + h_{n-1}x^{n-1} + \dots + h_0$ , where the term  $h_ix^i$  refers to the  $i$ th flip-flop of the register, such that, if  $h_i = 1$ , then there is a feedback tap taken from this flip-flop. Also,  $h_0 = 1$ . Fig. 1A shows a type I LFSR, or Fibonacci generator. Fig. 1B shows a type II LFSR, or Galois generator, which uses a shift register with interspersed XOR gates. If an LFSR (of either type) is initialized to a nonzero value, then it can cycle through a number of states before coming back to the initial state. A characteristic polynomial which causes an  $n$ -bit LFSR to go through all possible  $2^n - 1$  nonzero states is called a primitive characteristic polynomial. The corresponding LFSR is often referred to as a maximum-length  
30 LFSR, and the resultant output sequence is termed a maximum-length sequence or  $m$ -sequence.

[0005] An LHCA is a collection of memory cells connected in such a way that each cell is restricted to local neighborhood interactions. These relationships are expressed by rules that determine the next state of a given cell based on information received from its neighbors. For example, if cell  $c$  can communicate only with its two neighbors,  $c-1$  and  $c+1$ , the so-called rules 90 and 150 are usually employed. The rule 90 can be implemented by a linear logic according to the formula  
35  $x_c(t+1) = x_{c-1}(t) \oplus x_{c+1}(t)$ , while the rule 150 satisfies the equation  $x_c(t+1) = x_{c-1}(t) \oplus x_c(t) \oplus x_{c+1}(t)$ , where  $x_c(t)$  represents the state of cell  $c$  at time  $t$ . An example of an LHCA is shown in Fig. 1C. This automaton features null boundary conditions. That is, the conditions behave as if the boundary always supplies a zero from beyond the automaton to the inputs to the exterior cells. In an alternative embodiment, the LHCA has cyclic boundary conditions in which the inputs to the exterior cells are connected so that the automaton forms a circle. Contrary to the LHCA with null boundary conditions, the LHCA  
40 with cyclic boundary conditions is unable to produce an  $m$ -sequence.

[0006] Although LFSRs and LHCAs can generate a large set of pseudo-random test patterns from one seed, this set seldom provides sufficient fault coverage for a circuit-under-test. At best, 95-96% coverage of stuck-at faults can be achieved, even if test points are added to the circuit-under-test to address random pattern-resistant faults. If higher fault coverage is desired, the pseudo-random test patterns must be supplemented in some way. One supplementing technique is to provide from an external tester additional seeds to the PRPG that target specific faults not detected by the initial seed. Each additional seed generates a set of patterns. Each set, however, takes considerable time to generate. A compression and decompression method based on reseeding an LFSR with variable-length seeds is described in N. Zacharia, et al., "Decompression of Test Data Using Variable-Length Seed LFSRs," Proc. VLSI Test Symposium, pp. 426-433 (1995) and J. Rajski et al., "Test Data Decompression for Multiple Scan Designs with Boundary Scan", IEEE Trans. on Computers, vol. C-47, pp. 1188-1200 (1998). Another supplementing technique is to provide fully specified deterministic patterns that bypass the PRPG and target directly the remaining, random pattern resistant faults. This reduces testing time, but increases memory requirements because the external tester memory required to store these "top-up" patterns is significant, often exceeding 50% of the memory required for a complete set of deterministic patterns. United States Patent No. 5,694,402, for example, mentions generating pseudorandom and deterministic test patterns  
55 as part of a method for structurally testing an integrated circuit. Another alternative is described in L.F.C. Lew Yan Voon, "BIST Linear Generator Based on Complemented Outputs," Proceedings of the 10th IEEE VLSI Test Symposium, pp. 137-142 (1992) and involves an autonomous linear finite state machine with both complemented and uncomplemented register outputs.

**SUMMARY**

[0007] In accordance with the invention, a method for applying test patterns to scan chains in a circuit-under-test is described and shown herein. The method comprises, in a pseudo-random phase of operation, providing an initial value; generating from the initial value a set of pseudo-random test patterns; and applying the pseudo-random test patterns to the scan chains in the circuit-under-test. In a deterministic phase of operation, the method comprises providing a set of compressed deterministic test patterns; decompressing the compressed deterministic test patterns into decompressed deterministic test patterns; and applying the decompressed deterministic test patterns to the scan chains in the circuit-under-test. The decompressing of a compressed deterministic test pattern into a decompressed test pattern of bits occurs as the compressed deterministic test pattern is being provided. The applying of a decompressed deterministic test pattern to scan chains of the circuit-under-test may occur as a compressed deterministic test pattern is being provided.

[0008] In one aspect of the invention, the method may be applied to a circuit comprising a decompressor/PRPG, control circuitry, circuit logic, and scan chains. The control circuitry is coupled to the decompressor/PRPG and operable to cause the decompressor/PRPG to generate, in a pseudo-random phase of operation, a set of pseudo-random patterns and to generate, in a deterministic phase of operation, a set of deterministic test patterns. The scan chains are coupled to the circuit logic and operable to receive test patterns generated by the decompressor/PRPG and to capture responses to the test patterns generated by the circuit logic.

[0009] The decompressor/PRPG may include a linear finite state machine that can take various forms, such as a linear feedback shift register or a cellular automaton. The decompressor/PRPG may also include a phase shifter, which may be constructed of linear logic gates.

[0010] These and other aspects of the invention will become apparent from the following detailed description of an exemplary embodiment, which makes references to the accompanying drawings.

**BRIEF DESCRIPTION OF THE DRAWINGS****[0011]**

Figs. 1A-C are block diagrams of several well-known types of linear finite state machines that are used to generate pseudo-random test patterns.

Fig. 2 is a block diagram of a test system according to the invention.

Fig. 3 is a block diagram of a circuit that includes a decompressor/PRPG in the form of an LFSM, a phase shifter and scan chains in accordance with the invention.

Fig. 4 shows structure for initializing a decompressor/PRPG.

Fig. 5 shows further detail of a decompressor/PRPG.

Fig. 6 shows the logical expressions for the bits stored in each scan cell in the scan chain of Fig. 3 produced by the decompressor/PRPG and phase shifter.

Figs. 7A-D illustrate alternative embodiments of the LFSM shown in Fig. 3.

Fig. 8 illustrates an embodiment of a 32-bit LFSM.

Fig. 9 illustrates an alternative embodiment of a phase shifter.

Fig. 10 illustrates the use of parallel-to-serial conversion for applying a compressed test pattern to the decompressor/PRPG.

**DETAILED DESCRIPTION**

[0012] In accordance with the invention, testing of electronic circuits that include scan chains is accomplished in two phases of operation: a pseudo-random phase, in which a decompressor/PRPG generates pseudo-random test patterns for the circuit-under-test, and a deterministic phase, in which the decompressor/PRPG generates deterministic test patterns for the circuit-under test. The two testing phases can be executed in either order and at the substantially the same time (e.g., while the circuit being tested is connected to a tester) or at substantially different times. For example, the deterministic phase may be executed only during manufacture testing if desired, and the pseudo-random phase executed only during field testing. Or both phases can be executed at manufacture testing, and the pseudo-random-phase again executed periodically in the field to ensure that the circuit is maintained. By providing both types of test patterns and by providing the deterministic test patterns in a compressed format, a high level of fault coverage can be achieved with acceptable tester time and memory requirements. In an exemplary embodiment of the invention described herein, the decompressor/PRPG is embedded on a microchip that includes the circuit-under-test and executes the two phases of operation under the joint control of an on-chip BIST controller and an external tester.

[0013] Prior to testing, an automatic test pattern generation (ATPG) tool is employed to determine which deterministic test patterns are required to supplement the coverage of the pseudo-random patterns provided by the PRPG. This tool

performs fault simulation of all pseudo-random patterns applied to a circuit-under-test and determines thereby which faults are detected and which are not. The faults that are undetected by pseudo-random patterns are then targeted, one by one. For one or more testable faults, the ATPG tool produces a test cube - a partially specified test pattern. For each test cube another tool, a solver of systems of linear equations, may be used to produce a compressed deterministic test pattern. These compressed patterns, when applied to the inputs of the decompressor/PRPG and shifted into the scan chains, produce values consistent with all specified positions of the original test cube.

[0014] Fig. 2 is a block diagram of a test system 30 according to the invention. The system includes a tester 21 such as external automatic testing equipment (ATE) and a circuit 34 that includes as all or part of it a circuit-under-test (CUT) 24. Embedded within the CUT 24 is circuit logic to be tested and multiple scan chains 26. The circuit 34 also includes a decompressor/PRPG 36 coupled to the scan chains 26, control circuitry such as a BIST controller 25 and a set of one or more AND logic gates 27; a multiplexer 29; and a multiple input signature register (MISR) 42.

[0015] The tester 21 is operable to provide a set of compressed deterministic test patterns to the decompressor/PRPG 36 through tester scan channels 40 and the set of AND gates 27. The BIST controller 25 is operable to provide an initial value to the decompressor/ PRPG through a seed/reset line 38. (Alternatively, the tester 21 could be configured to provide this initial value over certain channels.) As will be explained, the control circuitry (the BIST controller and AND gates in this embodiment) are operable to configure the decompressor/PRPG to generate, in a pseudo-random phase of operation, a set of pseudo-random patterns from the initial value for the scan chains and to generate, in a deterministic phase of operation, a set of decompressed deterministic test patterns from the set of compressed deterministic test patterns for the scan chains. Other connections of particular interest in test system 30 include a tester-BIST controller communication line 35 between the controller 25 and the tester 21; a MISR reset select line 33 between the tester 21 and select terminal of the multiplexer 29; and a decompressor reset line 39 between the tester 21 and the decompressor/ PRPG 36. Connected to the input terminals of the multiplexer 29 are MISR reset lines 41 and 46 from the tester and BIST controller, respectively. The input terminal of the MISR 42 is coupled to the output terminal of the multiplexer 29, which thus passes a reset signal from either the tester or BIST controller to the MISR. The output terminal of the MISR is coupled to a register 43 within the tester 21, which receives the test response signatures from the MISR for comparison against reference signatures 44 at comparator 45. A decision (Go/No Go) is made based on the comparison whether the CUT 24 contains a fault.

[0016] It should be understood that this block diagram represents the functional relationships between the various parts of the test system. The physical design of the test system can vary, depending upon the desired implementation. Conventional elements of such a design, which are not shown, include clock signals, gates, power, etc. and other elements known in the art for carrying out the various functions.

[0017] Before describing the structure of the decompressor/PRPG 36 in more detail, the operation of the test system overall and the decompressor/PRPG will first be explained. Assume that the pseudo-random phase of testing is executed first, although the two phases of operation can be executed in either order. The tester 21 notifies the BIST controller via line 35 to initiate the generation of pseudo-random test patterns. Through line 33, the tester also disables AND gates 27 and selects the reset signal on line 46 as the output of multiplexer 29. By disabling the AND gates, the compressed, deterministic test patterns 32 from the tester are prevented from influencing the output of the decompressor/PRPG 36. And by selecting the reset signal on line 46, the tester enables controller 25 to reset the MISR before it stores compacted test responses from the scan chains.

[0018] The BIST controller 25 responds to the tester notification by transmitting a reset signal on line 38 (reset/seed) to the decompressor/PRPG and a reset signal on line 46 to the MISR 42. In the present embodiment, the initial value (seed) is hard-wired into the decompressor/PRPG, so that a high logic level on line 38 loads the seed into the decompressor/PRPG and a low logic level allows the decompressor/PRPG to generate pseudo-random patterns therefrom. Otherways for resetting and loading the seed are, of course, possible. Once the seed has been loaded, the decompressor/ PRPG is clocked to generate a set of pseudo-random test patterns that are applied to the scan chains 26 within the CUT 24 (e.g., two hundred fifty thousand patterns, with perhaps one hundred shifts per pattern to load each into the scan chains). After the circuit logic within the CUT 24 is clocked with each pseudo-random test pattern, the test response to that pattern is captured in the scan chains 26 and passed to the MISR 42, which compresses the multiple test pattern responses into a signature. This signature is then sent to the register 43 within the tester 21 for comparison with a reference signature 44 at comparator 45. If the comparison shows a difference, then a fault exists in CUT 24. The comparison technique can vary. For example, the pseudo-random test patterns and deterministic patterns may be compressed and compared separately to different reference signatures, as just described. Another option is to compress all the test patterns, pseudo-random and deterministic, into one signature and then compare it to a single reference signature.

[0019] The BIST controller 25 monitors the number of pseudo-random patterns applied and the shifts through the scan chains by counting the clock pulses controlling the decompressor/PRPG. When the clock pulses indicate that a desired number of the pseudo-random test patterns has been applied and the test results captured in the MISR, the controller notifies the tester 21 via line 35 that the pseudo-random phase of operation is complete. The tester responds

by initializing the test system 30 for the deterministic phase. This includes enabling the AND gates 27 via line 33 and selecting the tester MISR reset line 41 as the output of multiplexer 29. The tester also resets the decompressor/PRPG via line 39 in advance of applying the compressed deterministic patterns 32 from its memory and resets the MISR 42 via line 41, if two separate response signatures are generated.

[0020] The tester 21 then provides from a set of compressed test patterns 32 of bits, one pattern at a time, through scan channels 40 to the circuit 34. A compressed pattern, as will be described, contains far fewer bits than a fully specified (i.e., uncompressed or conventional) test pattern. A compressed pattern need contain only enough information to recreate deterministically specified bits, whereas a conventional pattern also contains randomly specified bits. Consequently, a compressed pattern is typically 2% to 5% of the size of a conventional test pattern and requires much less tester memory for storage than conventional patterns. As importantly, compressed test patterns require much less time to transmit from a tester to a CUT 24. (Alternatively, uncompressed deterministic test patterns can be used in addition to the compressed patterns, if desired.)

[0021] The compressed test patterns 32 are continuously provided from the tester 21 to scan chains 26 within the CUT 24 without interruption (i.e., the decompressing of a compressed deterministic test pattern into a decompressed test pattern of bits occurs as the compressed deterministic test pattern is being provided). Specifically, as a compressed test pattern is being provided by the tester 21 to the input channels of the decompressor/PRPG 36, it decompresses the compressed pattern into a decompressed pattern of bits. The decompressed test pattern is then applied to the scan chains 26 in the CUT 24, and can be done while the compressed test pattern is being provided to the circuit 34 (i.e., the applying of a decompressed deterministic test pattern to scan chains of the circuit-under-test can occur as a compressed deterministic test pattern is being provided). After the circuit logic within the CUT 24 is clocked with a decompressed deterministic test pattern, the test response to that pattern is captured in the scan chains and passed to the MISR 42, where it is compacted as part of a signature. When the response to all of the deterministic patterns have been compressed as a signature, the tester is notified and it requests that the signature be transmitted for comparison with the reference signature 44 at comparator 45. The comparison yields a test result (Go/No Go) that indicates whether the CUT 24 has a fault and should be rejected. As noted above, there may be a single signature or multiple partial signatures for a CUT 24 which are created and compared, depending on design choices.

[0022] In a typical configuration, the decompressor/PRPG 36 has one output per scan chain 26, and there are more scan chains than scan channels 40 to the decompressor/PRPG. However, as will be described, other configurations are also possible in which the decompressor/PRPG outputs are fewer than or equal to the input channels. The decompressor/PRPG generates in a given time period a greater number of decompressed bits at its outputs than the number of compressed pattern bits it receives during the same time period. This is an act of decompression, whereby the decompressor/PRPG 36 generates a greater number of bits than are provided to it in a given time period. As explained below, decompression comprises generating one or more bits of a decompressed pattern by logically combining two or more bits of a compressed test pattern. For example, the compressed bits can be combined with an XOR operation or an XNOR operation.

[0023] To reduce the data volume of the test response and the time for sending the response to the tester, the circuit 34 may include means for compressing the test response that is read from the scan chains 26. One structure for providing such compression is the multiple input signature register (MISR) 42 described above. Another structure for providing such compression is one or more spatial compactors (not shown), especially for the deterministic test patterns. The compressed test responses produced by the compactors are compared one by one with compressed reference responses within the tester. A fault is detected if a reference response does not match an actual response.

[0024] The providing of a compressed test pattern to a circuit, its decompression into a decompressed test pattern, and the application of the decompressed test pattern to the scan chains is performed synchronously, continuously, and substantially concurrently. The rate at which each act occurs, however, can vary. All acts can be performed synchronously at a same clock rate if desired. Or the acts can be performed at different clock rates. If the acts are performed at the same clock rate, or if the compressed test patterns are provided and decompressed at a higher clock rate than at which the decompressed test patterns are applied to the scan chains, then the number of outputs of decompressor/PRPG 36 and associated scan chains will exceed the number of input channels of the decompressor/PRPG. In this first case, decompression is achieved by providing more decompressor/PRPG outputs than input channels. If the compressed test patterns are provided at a lower clock rate and decompressed and applied to the scan chains at a higher clock rate, then the number of outputs and associated scan chains can be the same, fewer, or greater than the number of input channels. In this second case, decompression is achieved by generating the decompressed test pattern bits at a higher clock rate than the clock rate at which the compressed test pattern bits are provided.

[0025] Fig. 3 is a block diagram of one possible embodiment of a decompressor/PRPG 36 in accordance with the invention. The decompressor/PRPG 36 includes a number of input channels 37 through which compressed deterministic test patterns are received from the tester 21 via AND gates 27. The decompressor is also connected to lines 38 and 39 (shown in more detail in Fig. 4). In this embodiment, the decompressor/PRPG is a linear finite state machine (LFSM) 46 coupled, if desired, through its taps 48 to a phase shifter 50. The LFSM through the phase shifter provides highly

linearly independent test patterns to the inputs of numerous scan chains 26 in the CUT 24. The LFSM can be built on the basis of the canonical forms of linear feedback shift registers, cellular automata, or transformed LFSRs that can be obtained by applying a number of  $m$ -sequence preserving transformations. The output of the LFSM is applied through output channels to the phase shifter 50, which ensures that the decompressed pattern bits present within each of the multiple scan chains 26 at any given time do not overlap in pattern (i.e., are out of phase).

[0026] The concept of continuous flow decompression described herein rests on the fact noted above that deterministic test patterns typically have only between 2 to 5 % of bits deterministically specified, with the remaining bits randomly filled during test pattern generation. (Test patterns with partially specified bit positions are called test cubes, an example of which appears in Table 2.) These partially specified test cubes are compressed so that the test data volume that has to be stored externally is significantly reduced. The fewer the number of specified bits in a test cube, the better is the ability to encode the information into a compressed pattern. The ability to encode test cubes into a compressed pattern is exploited by having a few decompressor input channels driving the circuit-under-test, which are viewed by the tester as virtual scan chains. The actual CUT 24, however, has its memory elements connected into a large number of real scan chains. Under these circumstances, even a low-cost tester that has few scan channels and sufficiently small memory for storing test data can drive the circuit externally.

[0027] Fig. 4 shows one possible means for the tester and BIST controller to initialize the decompressor/PRPG 36. Line 39, which provides a tester reset signal before the application of each deterministic pattern, is coupled to each of the memory elements (stages) of the LFSM 46. When active (high), the signal on line 39 resets each memory element to a zero value through an AND gate 31 coupled to the input terminal of each element. When inactive (low), the reset signal has no effect on the operation of the LFSM. Line 38, which provides a seed/reset signal from the BIST controller, is also coupled to each memory element of the LFSM. The particular coupling at each element determines the binary value of the seed at that element; thus, the seed is "hard-wired" into the LFSM. When the signal on line 38 is high, it loads the seed by setting each element to a binary predetermined value. In Fig. 4, for example, element 7 is set to 0 because the high signal on line 38 disable the adjacent AND gate 31. Element 0, in contrast, is set to 1 because the OR gate 47 adjacent to the element passes the high signal to the element regardless of the value provided by the output terminal of adjacent element 1. Other initialization means are, of course, possible, such as transmitting a seed from the BIST controller or tester rather than wiring it into the LFSM.

[0028] Fig. 5 shows in more detail an exemplary decompressor/PRPG, with an LFSM embodied in an eight-stage type 1 LFSR 52 implementing primitive polynomial  $h(x) = x^8 + x^4 + x^3 + x^2 + 1$ . Lines 38 and 39 are shown connected to each memory element of the LFSR. The phase shifter 50, embodied in a number of XOR gates, drives eight scan chains 26, each eight bits long. The structure of the phase shifter is selected in such a way that a mutual separation between the shifter's output channels C0-C7 is at least eight bits, and all output channels are driven by 3-input (tap) XOR functions having the following forms:

Table 1

$$\begin{array}{ll} C_0 = s_4 \oplus s_3 \oplus s_0 & C_4 = s_4 \oplus s_2 \oplus s_1 \\ C_1 = s_7 \oplus s_6 \oplus s_5 & C_5 = s_5 \oplus s_2 \oplus s_0 \\ C_2 = s_7 \oplus s_3 \oplus s_2 & C_6 = s_6 \oplus s_5 \oplus s_3 \\ C_3 = s_6 \oplus s_1 \oplus s_0 & C_7 = s_7 \oplus s_2 \oplus s_0 \end{array}$$

where  $C_i$  is the  $i$ th output channel and  $s_k$  indicates the  $k$ th stage of the LFSR. Assume that the LFSR 52 is fed every clock cycle through its input channels 37a, 37b and input injectors 48a, 48b (XOR gates) to the second and the sixth stages of the register. The input variables "a" (compressed test pattern bits) received on channel 37a are labeled with even subscripts ( $a_0, a_2, a_4, \dots$ ) and the variables "a" received on channel 37b are labeled with odd subscripts ( $a_1, a_3, a_5, \dots$ ). Treating these external variables as Boolean, all scan cells can be conceptually filled with symbolic expressions being linear functions of input variables injected by tester 21 into the LFSR 52. Given the feedback polynomial, the phase shifter 50, the location of injectors 48a, b as well as an additional initial period of four clock cycles during which only the LFSR is supplied by test data, the contents of each scan cell within the scan chains 26 can be logically determined.

[0029] Fig. 6 gives the expressions for the sixty-four cells in the scan chains of Fig. 5, with the scan chains identified as 0 through 7 at the bottom of Fig. 5. The expressions for each scan chain cell in Fig. 6 are listed in the order in which the information is shifted into the chain, i.e., the topmost expression represents the data shifted in first to the chain and thus corresponds to the information stored in the bottom cell of that chain.

[0030] Assume that the decompressor/PRPG 36 in Fig. 5 is to generate a test pattern based on the following partially specified test cube in Table 2 (the contents of the eight scan chains are shown here horizontally, with the leftmost column representing the information that is shifted first into the scan chains and which thus resides in the bottom scan chain cell):

Table 2

5	xxxxxx	scan chain 0
10	xxxxxx	scan chain 1
	xxx11xx	scan chain 2
	xx0xx1x	scan chain 3
	xxx0xx1	scan chain 4
	xx0x0xx	scan chain 5
	xx1x1xx	scan chain 6
	xxxxxx	scan chain 7

There are only ten deterministically specified bits; the variable x denotes a "don't care" condition. Then a corresponding compressed test pattern can be determined by solving the following system of ten equations from Fig. 5 using any of a number of well-known techniques such as Gauss-Jordan elimination techniques. The selected equations correspond to the ten deterministically specified bits:

Table 3

$$\begin{aligned}
 & a_2 \oplus a_6 \oplus a_{11} = 1 \\
 & a_0 \oplus a_1 \oplus a_4 \oplus a_8 \oplus a_{13} = 1 \\
 & a_4 \oplus a_5 \oplus a_9 \oplus a_{11} = 0 \\
 & a_0 \oplus a_2 \oplus a_5 \oplus a_{12} \oplus a_{13} \oplus a_{17} \oplus a_{19} = 1 \\
 & a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_8 \oplus a_{12} \oplus a_{15} = 0 \\
 & a_0 \oplus a_1 \oplus a_3 \oplus a_5 \oplus a_7 \oplus a_8 \oplus a_{10} \oplus a_{11} \oplus a_{12} \oplus a_{14} \oplus a_{18} \oplus a_{21} = 1 \\
 & a_2 \oplus a_3 \oplus a_4 \oplus a_9 \oplus a_{10} = 0 \\
 & a_0 \oplus a_1 \oplus a_2 \oplus a_6 \oplus a_7 \oplus a_8 \oplus a_{13} \oplus a_{14} = 0 \\
 & a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_{10} = 1 \\
 & a_0 \oplus a_1 \oplus a_3 \oplus a_7 \oplus a_8 \oplus a_9 \oplus a_{10} \oplus a_{14} = 1
 \end{aligned}$$

It can be verified that the resulting seed variables  $a_0, a_1, a_2, a_3$  and  $a_{13}$  are equal to the value of one while the remaining variables assume the value of zero. This seed will subsequently produce a fully specified test pattern in the following form (the initial specified positions are underlined):

Table 4

35	10100100
40	11000100
	1 <u>11111110</u>
	0 <u>0010011</u>
	1 <u>0100001</u>
	1 <u>1010000</u>
	1 <u>1111111</u>
45	01001100

The compression ratio is defined as the number of scan cells divided by the number of bits required to place the test pattern within the cells. In the embodiment of Fig. 5, four bits are provided at each of the two injectors to fill the decompressor/PRPG and then eight additional bits are provided at each of the two injectors to complete the generation of content for each of the scan cells. This achieves a compression ratio of  $64 / (2 \times 8 + 2 \times 4) \approx 2.66$ . With no compression, the ratio would be less than one ( $64 / (8 \times 8 + 8 \times 4) \approx 0.67$ ).

[0031] Figs. 7A-D illustrate various embodiments for the LFSM 46 of Fig. 3. Fig. 7A is a type I LFSR 60. Fig. 7B is a type II LFSR 62. Fig 7C is a transformed LFSR 64. And Fig. 7D is a cellular automaton 66. All of them implement primitive polynomials. Except for the cellular automaton 66, in each case the LFSM includes a number of memory elements connected in a shift register configuration. In addition, there are several feedback connections between various memory cells that uniquely determine the next state of the LFSM. The feedback connections are assimilated into the design by introducing injectors in the form of XOR gates near the destination memory elements. The input channels 37a, 37b provide the bits of a compressed pattern to the LFSM through input injectors 48a, b. The injectors are handled similarly as the other feedback connections within the LFSM except that their sources of bits are the input channels. The input

channels 37 may have multiple fan-outs driving different LFSM injectors 48 to improve the encoding efficiency.

[0032] Fig. 8 shows a preferred embodiment of a 32-bit LFSM in the form of a re-timed LFSR 68. The injectors are spaced equally so that the input variables are distributed optimally once they are injected into the LFSM. In practice, the size of the LFSM depends on the number of real scan chains in a circuit, the desired compression ratio of encoding, and on certain structural properties of the circuit-under-test.

[0033] Fig. 9 illustrates an alternative embodiment of a phase shifter 50, constructed with an array of XNOR gates rather than XOR gates. Phase shifters can be constructed with combinations of XNOR and XOR gates as well.

[0034] Fig. 10 illustrates the use of parallel-to-serial conversion for applying a compressed test pattern to the decompressor. If the input channels 37 of the decompressor/PRPG 36 are fewer in number than the number of scan channels 40 of the tester 21, it can be advantageous to provide a parallel-to-serial converter such as registers 70 at the input to the decompressor. The registers 70 are clocked such that their contents are shifted out before the next set of bits is applied to the register from the tester 21. The continuous flow of the test patterns is thus preserved.

[0035] The process of decompressing a deterministic test pattern will now be described in more detail, with reference to Fig. 3. The LFSM 46 starts its operation from an initial all-zero state provided by a reset signal on line 39 by tester 21. Assuming an n-bit LFSM and m input injectors,  $\lceil n/m \rceil$  clock cycles may be used to initialize the LFSM before it starts generating bits corresponding to the actual test patterns. After initialization of the LFSM and assuming the appropriate system clocks are running at the same rate, a new bit is loaded in parallel into each scan chain 26 every clock cycle via the phase shifter 50. At this time, the circuit-under-test 24 is operated in the scan mode, so that the decompressed test pattern fills the scan chains 26 with 0s and 1s (and shifts out any previous test response stored there). A small number of bit positions in the scan chains, therefore, get deterministically specified values while the remaining positions are filled with random bits generated by the LFSM. The number of clock cycles for which a test pattern is shifted is determined by the length of the longest scan chain within the circuit, the number being at least as great as the number of cells in the longest scan chain. A scan-shift signal is therefore held high for all the scan chains until the longest scan chain gets the entire test pattern. The shorter scan chains in the circuit are left justified so that the first few bits that are shifted are overwritten without any loss of information.

[0036] Patterns from the LFSM may be linearly dependent. In other words, it is possible to determine various bit positions within the two-dimensional structure of multiple scan chains that are significantly correlated. This causes testability problems, as it is often not possible to provide the necessary stimulus for fault excitation to the gates driven by positions that have some form of dependency between them. Consequently, the phase shifter 50 (such as an array of XOR gates or XNOR gates) may be employed at the taps (outputs) of the LFSM to reduce linear dependencies between various bit positions within the scan chains. The XOR logic can be two-level or multilevel depending on the size of the XOR gates. Every scan chain in the CUT 24 is driven by signals that are obtained by XOR-ing a subset of taps 48 from the LFSM. These taps are determined so that the encoding efficiency of the test cubes is still preserved. In addition, the taps are selected in a manner so that all memory cells in the LFSM have approximately equal number of fan-out signals and the propagation delays are suitably optimized. Once a decompressed test pattern is completely loaded into the scan chains during test mode, the CUT 24 is switched to the normal mode of operation. The CUT then performs its normal operation under the stimulus provided by the test pattern in the scan chains. The test response of the CUT is captured in the scan chains. During the capture the LFSM is reset to the all-zero state before a new initialization cycle begins for loading the next test pattern.

[0037] Having illustrated and described the principles of the invention in exemplary embodiments, it should be apparent to those skilled in the art that the illustrative embodiments can be modified in arrangement and detail without departing from such principles. For example, the order of steps in the various methods can be varied. In view of the many possible embodiments to which the principles of the invention may be applied, it should be understood that the illustrative embodiment is intended to teach these principles and is not intended to be a limitation on the scope of the invention, as defined by the appended claims.

## Claims

50 1. A method for applying test patterns to scan chains in a circuit-under-test, the method comprising:

in a pseudo-random phase of operation:

providing an initial value;

generating from the initial value a set of pseudo-random test patterns; and

applying the pseudo-random test patterns to the scan chains in the circuit-under-test;

in a deterministic phase of operation:

providing a set of compressed deterministic test patterns;  
 decompressing a compressed deterministic test pattern into a decompressed deterministic test pattern as the compressed deterministic test pattern is being provided; and  
 applying the decompressed deterministic test patterns to the scan chains in the circuit-under-test.

- 5        2. The method of claim 1 wherein the pseudo-random phase executes before the deterministic phase.
- 10      3. The method of claim 1 wherein the deterministic phase executes before the pseudo-random phase.
- 15      4. The method of claim 1 wherein the deterministic phase and the pseudo-random phase are executed while the circuit-under-test is connected to a tester.
- 20      5. The method of claim 1 wherein the deterministic phase and the pseudo-random phase are executed at substantially different times.
- 25      6. The method of any of claims 1 through 5 wherein decompressing a compressed deterministic test pattern comprises generating one or more bits of a decompressed pattern by logically combining two or more bits of a compressed test pattern.
- 30      7. The method of claim 6 wherein logically combining two or more bits of the compressed test pattern comprises combining the bits with an XOR operation.
- 35      8. The method of claim 6 wherein logically combining two or more bits of the compressed test pattern comprises combining the bits with an XNOR operation.
- 40      9. The method of any of claims 1 through 8 wherein the applying of a decompressed deterministic test pattern to scan chains of the circuit-under-test occurs as a compressed deterministic test pattern is being provided.
- 45      10. The method of any of claims 1 through 9 wherein providing a compressed test pattern comprises generating a serial stream of bits representing the compressed test pattern.
- 50      11. The method of any of claims 1 through 9 wherein providing a compressed test pattern comprises generating a parallel stream of bits representing the compressed test pattern.
- 55      12. The method of claim any of claims 1 through 11 including:
  - capturing in the scan chains the results of test pattern applications; and
  - comparing the results with a reference value.
- 60      13. A circuit (34) comprising:
  - a decompressor/PRPG (36);
  - control circuitry coupled to the decompressor/PRPG (36) and operable to cause the decompressor/PRPG (36) to generate, in a pseudo-random phase of operation, a set of pseudo-random patterns and to generate, in a deterministic phase of operation, a set of decompressed deterministic test patterns from a set of provided compressed deterministic patterns, the control circuitry comprising one or more logic gates that receive the compressed deterministic patterns during the deterministic phase of operation and logically combine the compressed deterministic patterns with bits stored within the decompressor/PRPG (36);
  - circuit logic; and
  - scan chains (26) coupled to the circuit logic and operable to receive test patterns generated by the decompressor/PRPG (36) and to capture responses to the test patterns generated by the circuit logic,

wherein the decompressor/PRPG (36) is operable to decompress a compressed deterministic test pattern as the compressed deterministic test pattern is being provided to the decompressor/PRPG (36).
- 65      14. The circuit (34) of claim 13 wherein the decompressor/PRPG (36) includes a linear finite state machine.
- 70      15. The circuit (34) of claim 14 wherein the linear finite state machine includes a linear feedback shift register.

16. The circuit (34) of claim 14 wherein the linear finite state machine includes a cellular automaton.
17. The circuit (34) of any of claims 13 through 16 wherein the decompressor/PRPG (36) includes a phase shifter.
- 5    18. The circuit (34) of claim 17 wherein the phase shifter comprises an array of XOR gates.
19. The circuit (34) of claim 17 wherein the phase shifter comprises an array of XNOR gates.
- 10    20. The circuit (34) of any of claims 13 through 19 wherein the control circuitry is operable to provide an initial value to the decompressor/PRPG (36) from which the decompressor/PRPG (36) generates a set of pseudo-random test patterns.
21. The circuit (34) of claim 20 wherein the control circuitry includes a BIST controller.
- 15    22. The circuit (34) of any of claims 13 through 21 wherein the decompressor/PRPG (36) is operable to receive from an external tester (21) an initial value from which the decompressor/PRPG (36) generates a set of pseudo-random test patterns.
- 20    23. The circuit (34) of any of claims 13 through 22 wherein the control circuitry is operable to direct a set of compressed deterministic test patterns from an external tester (21) to the decompressor/PRPG (36) from which the decompressor/PRPG (36) generates a set of decompressed deterministic test patterns.
- 25    24. The circuit (34) of claim 23 wherein the one or more logic gates are logic gates through which the compressed deterministic patterns pass from the external tester to the decompressor/PRPG (36), the logic gates responsive to an enable signal.
- 25    25. The circuit (34) of any of claims 13 through 24 wherein the decompressor/PRPG (36) is operable to apply a decompressed deterministic test pattern to the scan chains (26) as the compressed deterministic test pattern is being provided to the decompressor/PRPG (36).
- 30    26. The circuit (34) of any of claims 13 through 25 including a multiple input signature register coupled to the scan chains (26) and operable to compress responses to test patterns captured by the scan chains (26).
- 35    27. The circuit (34) of claim 23, wherein the one or more logic gates are XOR gates.

### Patentansprüche

1. Ein Verfahren zum Anwenden von Prüfmustern auf Scanketten in einer zu prüfenden Schaltung, wobei das Verfahren umfasst:
  - in einer pseudozufälligen Betriebsphase:
    - das Bereitstellen eines Ausgangswerts;
    - 45    das Erzeugen einer Menge von pseudozufälligen Prüfmustern anhand des Ausgangswerts; und
    - das Anwenden der pseudozufälligen Prüfmuster auf die Scanketten in der zu prüfenden Schaltung;
  - 50    in einer deterministischen Betriebsphase:
    - das Bereitstellen einer Menge von komprimierten deterministischen Prüfmustern;
    - das Dekomprimieren eines komprimierten deterministischen Prüfmusters in ein dekomprimiertes deterministisches Prüfmuster im Zuge des Bereitstellens des komprimierten deterministischen Prüfmusters; und
    - das Anwenden des dekomprimierten deterministischen Prüfmusters auf die Scanketten in der zu prüfenden Schaltung.
- 55    2. Das Verfahren gemäß Anspruch 1, bei dem die pseudozufällige Phase vor der deterministischen Phase ausgeführt wird.

3. Das Verfahren gemäß Anspruch 1, bei dem die deterministische Phase vor der pseudozufälligen Phase ausgeführt wird.
4. Das Verfahren gemäß Anspruch 1, bei dem die deterministische Phase und die pseudozufällige Phase ausgeführt werden, während die zu prüfende Schaltung an ein Prüfgerät angeschlossen ist.
5. Das Verfahren gemäß Anspruch 1, bei dem die deterministische Phase und die pseudozufällige Phase zu im Wesentlichen unterschiedlichen Zeiten ausgeführt werden.
10. Das Verfahren gemäß einem der Ansprüche 1 bis 5, bei dem das Dekomprimieren eines komprimierten deterministischen Prüfmusters das Erzeugen von einem oder mehr Bits eines dekomprimierten Musters durch logisches Verknüpfen von zwei oder mehr Bits eines komprimierten Prüfmusters umfasst.
15. Das Verfahren gemäß Anspruch 6, bei dem das logische Verknüpfen von zwei oder mehr Bits des komprimierten Prüfmusters das Verknüpfen der Bits durch eine XOR-Verknüpfung umfasst.
20. Das Verfahren gemäß Anspruch 6, bei dem das logische Verknüpfen von zwei oder mehr Bits des komprimierten Prüfmusters das Verknüpfen der Bits durch eine XNOR-Verknüpfung umfasst.
25. Das Verfahren gemäß einem der Ansprüche 1 bis 9, bei dem das Anwenden eines dekomprimierten deterministischen Prüfmusters auf Scanketten der zu prüfenden Schaltung im Zuge des Bereitstellens eines komprimierten deterministischen Prüfmusters erfolgt.
30. Das Verfahren gemäß einem der Ansprüche 1 bis 11, das beinhaltet
  - das Erfassen der Ergebnisse der Anwendungen von Prüfmustern in den Scanketten; und
  - das Vergleichen der Ergebnisse mit einem Bezugswert.
35. Eine Schaltung (34), die umfasst:
  - ein Dekompressionsgerät/PRPG (36);
  - eine Steuerelektronik, die mit dem Dekompressionsgerät/PRPG (36) gekoppelt ist und betrieben werden kann, um das Dekompressionsgerät/PRPG (36) zu verlassen, in einer pseudozufälligen Betriebsphase eine Menge von pseudozufälligen Mustern zu erzeugen und in einer deterministischen Betriebsphase eine Menge von dekomprimierten deterministischen Prüfmustern anhand einer Menge von bereitgestellten komprimierten deterministischen Mustern zu erzeugen, wobei die Steuerelektronik ein oder mehrere logische Gatter umfasst, die während der deterministischen Betriebsphase die komprimierten deterministischen Muster empfangen und die komprimierten deterministischen Muster mit den im Dekompressionsgerät/PRPG (36) gespeicherten Bits logisch verknüpfen;
  - eine Schaltungslogik; und
  - Scanketten (26), die mit der Schaltungslogik gekoppelt sind und betrieben werden können, um vom Dekompressionsgerät/PRPG (36) erzeugte Prüfmuster zu empfangen und von der Schaltungslogik erzeugte Reaktionen auf die Prüfmuster zu erfassen,
40. wobei das Dekompressionsgerät/PRPG (36) betrieben werden kann, um ein komprimiertes deterministisches Prüfmuster im Zuge des Bereitstellens des komprimierten deterministischen Prüfmusters für das Dekompressionsgerät/PRPG (36) zu dekomprimieren.
45. Die Schaltung (34) gemäß Anspruch 13, bei der das Dekompressionsgerät/PRPG (36) einen linearen endlichen Zustandsautomaten beinhaltet.
50. Die Schaltung (34) gemäß Anspruch 14, bei der der lineare endliche Zustandsautomat ein lineares rückgekoppeltes
55. Die Schaltung (34) gemäß Anspruch 15, bei der der lineare endliche Zustandsautomat ein lineares rückgekoppeltes

Schieberegister beinhaltet.

16. Die Schaltung (34) gemäß Anspruch 14, bei der der lineare endliche Zustandsautomat einen Zellularautomaten beinhaltet.

5        17. Die Schaltung (34) gemäß einem der Ansprüche 13 bis 16, bei der das Dekompressionsgerät/PRPG (36) einen Phasenschieber beinhaltet.

10      18. Die Schaltung (34) gemäß Anspruch 17, bei der der Phasenschieber eine Anordnung von XOR-Gattern umfasst.

15      19. Die Schaltung (34) gemäß Anspruch 17, bei der der Phasenschieber eine Anordnung von XNOR-Gattern umfasst.  
20. Die Schaltung (34) gemäß einem der Ansprüche 13 bis 19, bei der die Steuerelektronik betrieben werden kann, um einen Ausgangswert für das Dekompressionsgerät/PRPG (36) bereitzustellen, anhand dessen das Dekompressionsgerät/PRPG (36) eine Menge von pseudozufälligen Prüfmustern erzeugt.

25      21. Die Schaltung (34) gemäß Anspruch 20, bei der die Steuerelektronik einen BIST-Controller beinhaltet.

20      22. Die Schaltung (34) gemäß einem der Ansprüche 13 bis 21, bei der das Dekompressionsgerät/PRPG (36) betrieben werden kann, um von einem externen Prüfgerät (21) einen Ausgangswert zu empfangen, anhand dessen das Dekompressionsgerät/PRPG (36) eine Menge von pseudozufälligen Prüfmustern erzeugt.

25      23. Die Schaltung (34) gemäß einem der Ansprüche 13 bis 22, bei der die Steuerelektronik betrieben werden kann, um eine Menge von komprimierten deterministischen Prüfmustern von einem externen Prüfgerät (31) an das Dekompressionsgerät/PRPG (36) zu leiten, anhand deren das Dekompressionsgerät/PRPG (36) eine Menge von dekomprimierten deterministischen Prüfmustern erzeugt.

30      24. Die Schaltung (34) gemäß Anspruch 23, bei der die ein oder mehreren logischen Gatter logische Gatter sind, die von den komprimierten deterministischen Mustern auf dem Weg vom externen Prüfgerät zum Dekompressionsgerät/PRPG (36) durchlaufen werden, wobei die logischen Gatter auf ein Freigabesignal reagieren.

35      25. Die Schaltung (34) gemäß einem der Ansprüche 13 bis 24, bei der das Dekompressionsgerät/PRPG (36) betrieben werden kann, um im Zuge des Bereitstellens des komprimierten deterministischen Prüfmusters für das Dekompressionsgerät/PRPG (36) ein dekomprimiertes deterministisches Prüfmuster auf die Scanketten (26) anzuwenden.

40      26. Die Schaltung (34) gemäß einem der Ansprüche 13 bis 25, die ein Signaturregister mit mehreren Eingängen beinhaltet, das mit den Scanketten (26) gekoppelt ist und betrieben werden kann, um von den Scanketten (26) erfassete Reaktionen auf Prüfmuster zu komprimieren.

45      27. Die Schaltung (34) gemäß Anspruch 23, bei der die eine oder mehreren logischen Gatter XOR-Gatter sind.

### Revendications

45      1. Un procédé pour appliquer des séquences de test à des chaînes de balayage dans un circuit sous test, le procédé comportant :

dans une phase d'opération pseudo-aléatoire :

50      fournir une valeur initiale ;  
générer à partir de la valeur initiale un ensemble de séquences de test pseudo-aléatoires ; et  
appliquer les séquences de test pseudo-aléatoires aux chaînes de balayage dans le circuit sous test ;

dans une phase d'opération déterministe :

55      fournir un ensemble de séquences de test déterministes compressées ;  
décompresser une séquence de test déterministe compressée en une séquence de test déterministe décompressée à mesure que la séquence de test déterministe compressée est fournie ; et

appliquer les séquences de test déterministes décompressées aux chaînes de balayage dans le circuit sous test.

2. Le procédé de la revendication 1 dans lequel la phase pseudo-aléatoire s'exécute avant la phase déterministe.

5 3. Le procédé de la revendication 1 dans lequel la phase déterministe s'exécute avant la phase pseudo-aléatoire.

4. Le procédé de la revendication 1 dans lequel la phase déterministe et la phase pseudo-aléatoire sont exécutées alors que le circuit sous test est connecté à un testeur.

10 5. Le procédé de la revendication 1 dans lequel la phase déterministe et la phase pseudo-aléatoire sont exécutées à des moments essentiellement différents.

15 6. Le procédé de n'importe lesquelles des revendications 1 à 5 dans lequel la décompression d'une séquence de test déterministe compressée comporte la génération d'un ou plusieurs bits d'une séquence décompressée en combinant logiquement deux bits ou plus d'une séquence de test compressée.

20 7. Le procédé de la revendication 6 dans lequel la combinaison logique de deux bits ou plus de la séquence de test compressée comporte la combinaison des bits avec une opération XOR.

25 8. Le procédé de la revendication 6 dans lequel la combinaison logique de deux bits ou plus de la séquence de test compressée comporte la combinaison des bits avec une opération XNOR.

9. Le procédé de n'importe lesquelles des revendications 1 à 8 dans lequel l'application d'une séquence de test déterministe décompressée à des chaînes de balayage du circuit sous test se produit à mesure qu'une séquence de test déterministe compressée est fournie.

30 10. Le procédé de n'importe lesquelles des revendications 1 à 9 dans lequel la fourniture d'une séquence de test compressée comporte la génération d'un train de bits série représentant la séquence de test compressée.

35 11. Le procédé de n'importe lesquelles des revendications 1 à 9 dans lequel la fourniture d'une séquence de test compressée comporte la génération d'un train de bits parallèle représentant la séquence de test compressée.

12. Le procédé de n'importe lesquelles des revendications 1 à 11, comprenant :

35 la capture dans les chaînes de balayage des résultats d'applications de séquence de test ; et la comparaison des résultats avec une valeur de référence.

13. Un circuit (34) comportant :

40 un décompresseur/générateur de séquences de test pseudo-aléatoires (36) ;  
une circuiterie de contrôle couplée au décompresseur/générateur de séquences de test pseudo-aléatoires (36) et utilisable pour amener le décompresseur/générateur de séquences de test pseudo-aléatoires (36) à générer, dans une phase d'opération pseudo-aléatoire, un ensemble de séquences pseudo-aléatoires et à générer, dans une phase d'opération déterministe, un ensemble de séquences de test déterministes décompressées à partir d'un ensemble de séquences déterministes compressées fournies, la circuiterie de contrôle comportant une ou plusieurs portes logiques qui reçoivent les séquences déterministes compressées durant la phase d'opération déterministe et combinent logiquement les séquences déterministes compressées avec des bits stockés à l'intérieur du décompresseur/générateur de séquences de test pseudo-aléatoires (36) ;  
une logique de circuit ; et

45 des chaînes de balayage (26) couplées à la logique de circuit et utilisables pour recevoir des séquences de test générées par le décompresseur/générateur de séquences de test pseudo-aléatoires (36) et pour capturer des réponses aux séquences de test générées par la logique de circuit,

50 dans lequel le décompresseur/générateur de séquences de test pseudo-aléatoires (36) est utilisable pour décompresser une séquence de test déterministe compressée à mesure que la séquence de test déterministe compressée est fournie au décompresseur/générateur de séquences de test pseudo-aléatoires (36).

14. Le circuit (34) de la revendication 13 dans lequel le décompresseur/générateur de séquences de test pseudo-aléatoires (36) comprend une machine à états finis linéaire.

5      15. Le circuit (34) de la revendication 14 dans lequel la machine à états finis linéaire comprend un registre à décalage à boucle fermée linéaire.

10     16. Le circuit (34) de la revendication 14 dans lequel la machine à états finis linéaire comprend un automate cellulaire.

10     17. Le circuit (34) de n'importe lesquelles des revendications 13 à 16 dans lequel le décompresseur/générateur de séquences de test pseudo-aléatoires (36) comprend un déphaseur.

15     18. Le circuit (34) de la revendication 17 dans lequel le déphaseur comporte un réseau de portes XOR.

15     19. Le circuit (34) de la revendication 17 dans lequel le déphaseur comporte un réseau de portes XNOR.

20     20. Le circuit (34) de n'importe lesquelles des revendications 13 à 19 dans lequel la circuiterie de contrôle est utilisable pour fournir une valeur initiale au décompresseur/générateur de séquences de test pseudo-aléatoires (36) à partir de laquelle le décompresseur/générateur de séquences de test pseudo-aléatoires (36) génère un ensemble de séquences de test pseudo-aléatoires.

20     21. Le circuit (34) de la revendication 20 dans lequel la circuiterie de contrôle comprend un contrôleur d'autotest intégré.

25     22. Le circuit (34) de n'importe lesquelles des revendications 13 à 21 dans lequel le décompresseur/générateur de séquences de test pseudo-aléatoires (36) est utilisable pour recevoir d'un testeur externe (21) une valeur initiale à partir de laquelle le décompresseur/générateur de séquences de test pseudo-aléatoires (36) génère un ensemble de séquences de test pseudo-aléatoires.

30     23. Le circuit (34) de n'importe lesquelles des revendications 13 à 22 dans lequel la circuiterie de contrôle est utilisable pour diriger un ensemble de séquences de test déterministes compressées d'un testeur externe (21) au décompresseur/générateur de séquences de test pseudo-aléatoires (36) à partir desquelles le décompresseur/générateur de séquences de test pseudo-aléatoires (36) génère un ensemble de séquences de test déterministes décompressées.

35     24. Le circuit (34) de la revendication 23 dans lequel la ou les portes logiques sont des portes logiques par lesquelles les séquences déterministes compressées passent du testeur externe au décompresseur/générateur de séquences de test pseudo-aléatoires (36), les portes logiques étant sensibles à un signal de validation.

40     25. Le circuit (34) de n'importe lesquelles des revendications 13 à 24 dans lequel le décompresseur/générateur de séquences de test pseudo-aléatoires (36) est utilisable pour appliquer une séquence de test déterministe décompressée aux chaînes de balayage (26) à mesure que la séquence de test déterministe compressée est fournie au décompresseur/générateur de séquences de test pseudo-aléatoires (36).

45     26. Le circuit (34) de n'importe lesquelles des revendications 13 à 25 comprenant un registre de signature à entrées multiples couple aux chaînes de balayage (26) et utilisable pour compresser les réponses aux séquences de test capturées par les chaînes de balayage (26).

45     27. Le circuit (34) de la revendication 23, dans lequel la ou les portes logiques sont des portes XOR.

50

55

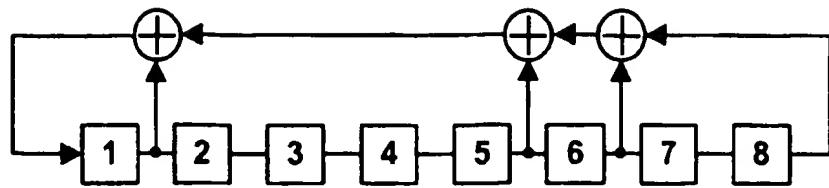


FIG. 1A

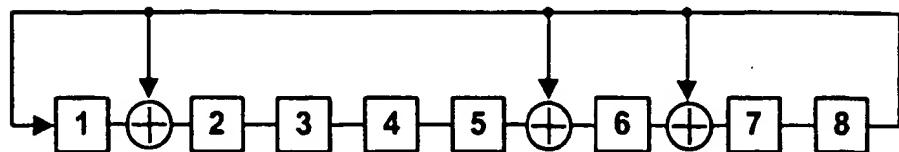


FIG. 1B

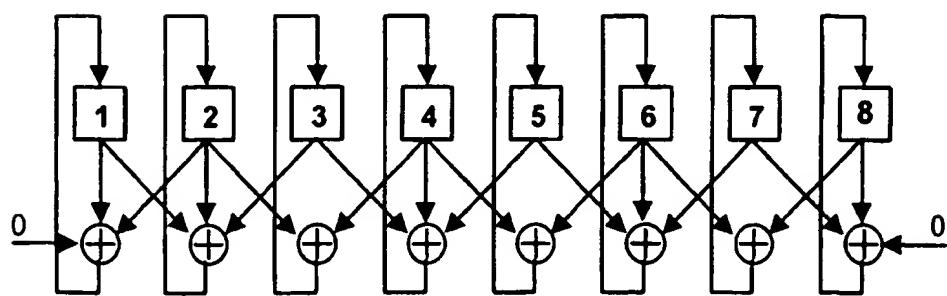


FIG. 1C

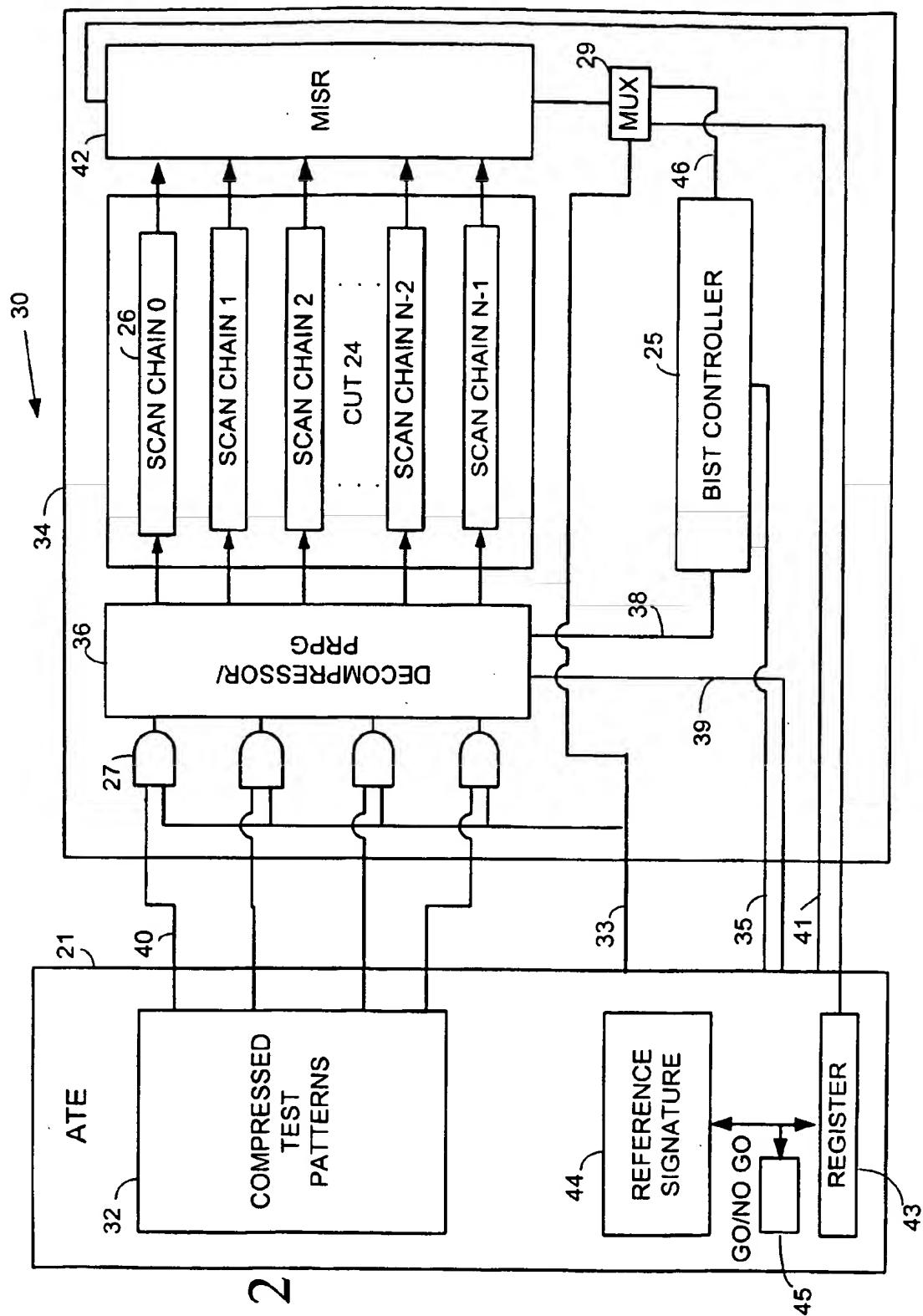


FIG. 2

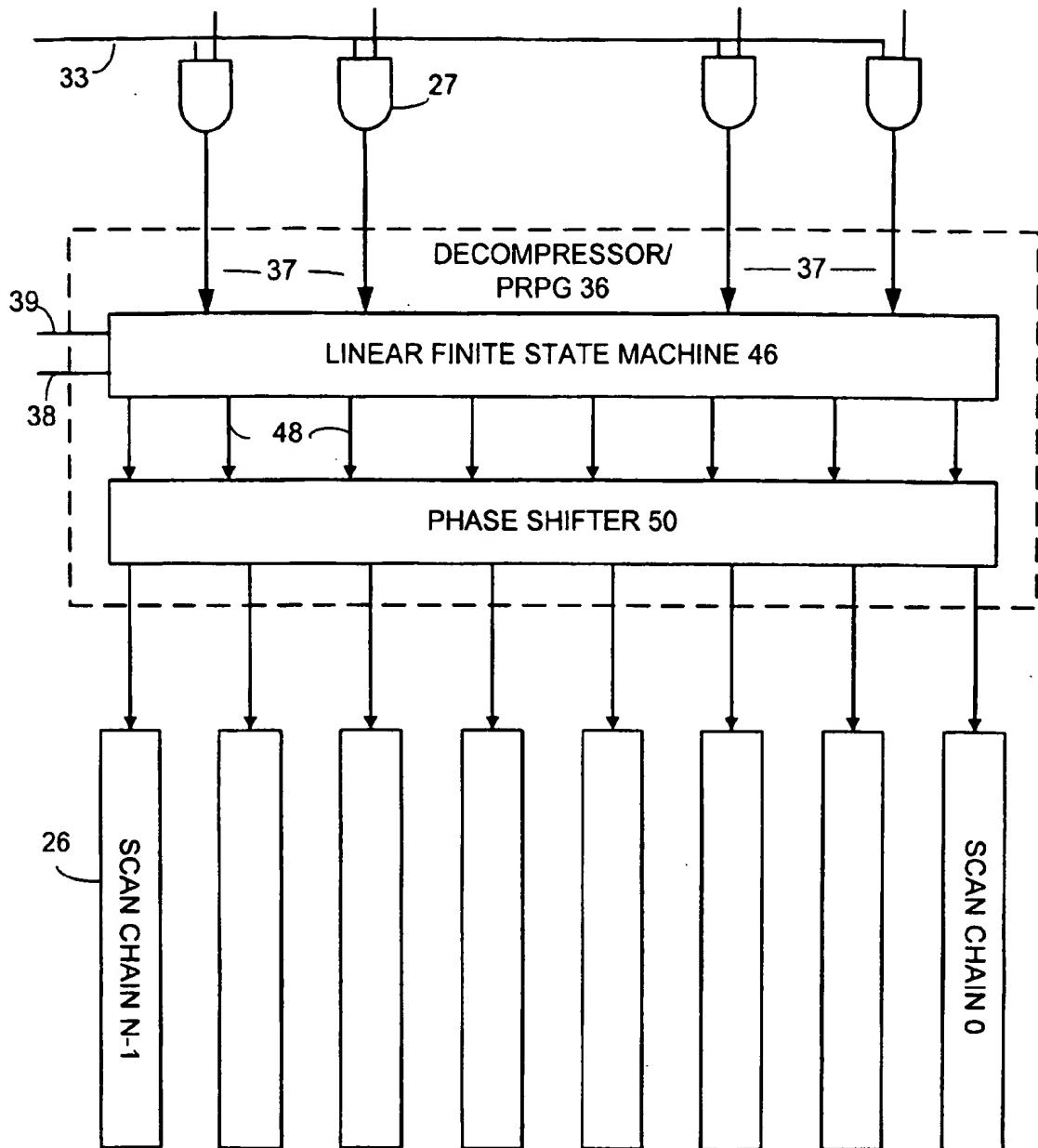


FIG. 3

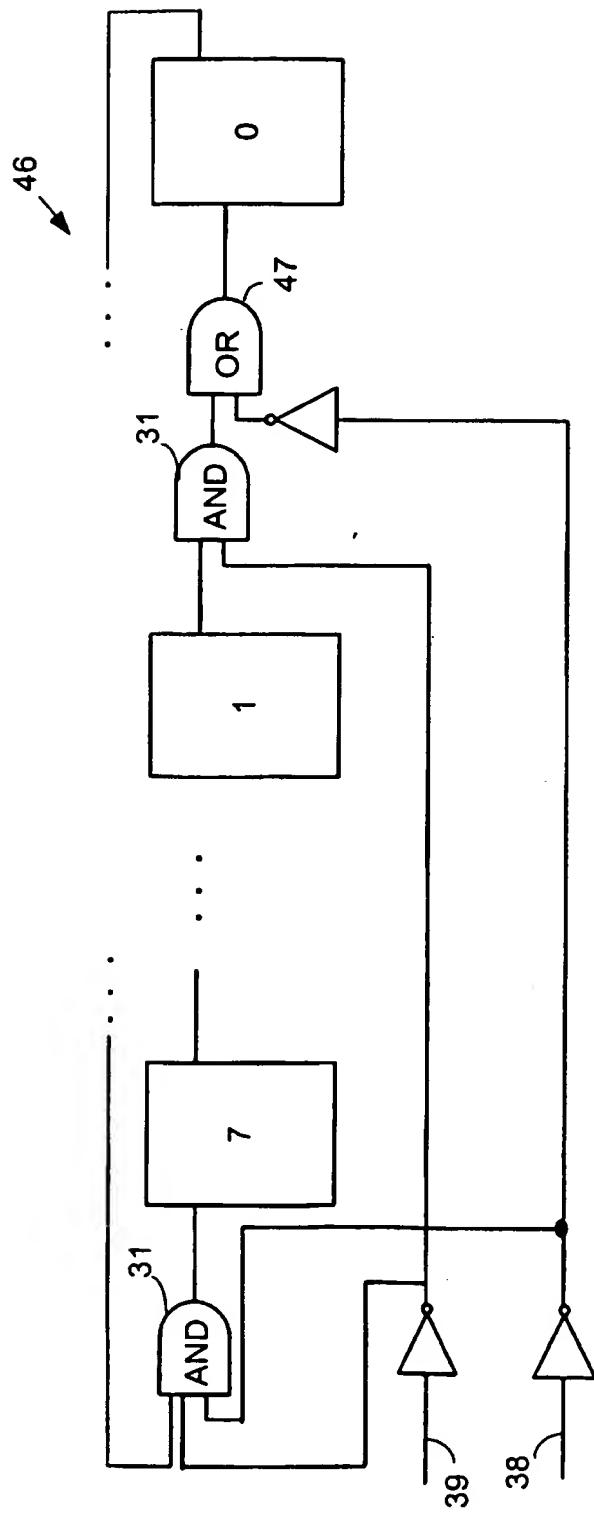


FIG. 4

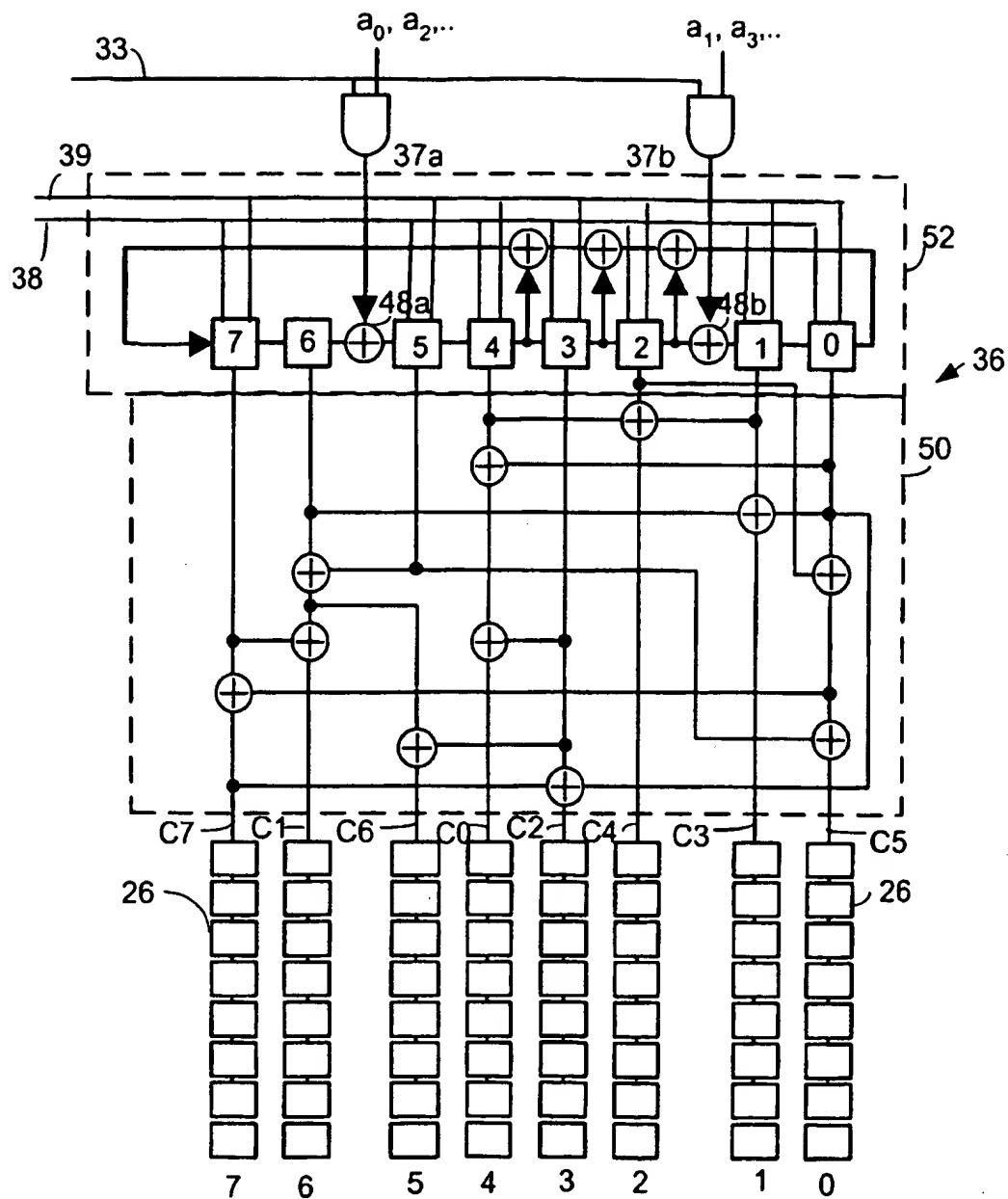


FIG. 5

## Scan chain 0

$$\begin{aligned}
 & a_2 \oplus a_4 \oplus a_3 \\
 & a_4 \oplus a_6 \oplus a_7 \\
 & a_1 \oplus a_6 \oplus a_8 \oplus a_9 \\
 & a_1 \oplus a_3 \oplus a_8 \oplus a_{10} \oplus a_{11} \\
 & a_3 \oplus a_5 \oplus a_{10} \oplus a_{12} \oplus a_{13} \\
 & a_0 \oplus a_1 \oplus a_4 \oplus a_8 \oplus a_9 \oplus a_{11} \oplus a_{12} \oplus a_{13} \oplus a_{16} \\
 & a_0 \oplus a_2 \oplus a_7 \oplus a_9 \oplus a_{14} \oplus a_{16} \oplus a_{17} \\
 & a_0 \oplus a_2 \oplus a_4 \oplus a_9 \oplus a_{11} \oplus a_{16} \oplus a_{18} \oplus a_{19}
 \end{aligned}$$

## Scan chain 2

$$\begin{aligned}
 & a_3 \\
 & a_0 \oplus a_5 \\
 & a_2 \oplus a_7 \\
 & a_0 \oplus a_4 \oplus a_9 \\
 & \underline{a_2 \oplus a_6 \oplus a_{11}} \\
 & a_0 \oplus a_1 \oplus a_4 \oplus a_8 \oplus a_{13} \\
 & a_0 \oplus a_2 \oplus a_3 \oplus a_6 \oplus a_{10} \oplus a_{15} \\
 & a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus a_8 \oplus a_{12} \oplus a_{17}
 \end{aligned}$$

## Scan chain 4

$$\begin{aligned}
 & a_0 \oplus a_4 \oplus a_7 \\
 & a_0 \oplus a_2 \oplus a_6 \oplus a_9 \\
 & a_0 \oplus a_1 \oplus a_2 \oplus a_4 \oplus a_8 \oplus a_{11} \\
 & a_0 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_6 \oplus a_{10} \oplus a_{13} \\
 & \underline{a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_8 \oplus a_{12} \oplus a_{15}} \\
 & a_1 \oplus a_3 \oplus a_4 \oplus a_6 \oplus a_7 \oplus a_8 \oplus a_{10} \oplus a_{14} \oplus a_{17} \\
 & a_1 \oplus a_3 \oplus a_4 \oplus a_6 \oplus a_8 \oplus a_9 \oplus a_{10} \oplus a_{12} \oplus a_{16} \oplus a_{19} \\
 & a_0 \oplus a_1 \oplus a_3 \oplus a_5 \oplus a_7 \oplus a_8 \oplus a_{10} \oplus a_{11} \oplus a_{12} \oplus a_{14} \oplus a_{18} \oplus a_{21}
 \end{aligned}$$

## Scan chain 6

$$\begin{aligned}
 & a_0 \oplus a_1 \oplus a_2 \oplus a_6 \\
 & a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_8 \\
 & a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_{10} \\
 & \underline{a_1 \oplus a_5 \oplus a_6 \oplus a_7 \oplus a_8 \oplus a_{12}} \\
 & a_0 \oplus a_1 \oplus a_3 \oplus a_7 \oplus a_8 \oplus a_9 \oplus a_{10} \oplus a_{14} \\
 & a_0 \oplus a_2 \oplus a_3 \oplus a_5 \oplus a_9 \oplus a_{10} \oplus a_{11} \oplus a_{12} \oplus a_{16} \\
 & a_0 \oplus a_2 \oplus a_4 \oplus a_5 \oplus a_7 \oplus a_{11} \oplus a_{12} \oplus a_{13} \oplus a_{14} \oplus a_{18} \\
 & a_0 \oplus a_2 \oplus a_4 \oplus a_6 \oplus a_7 \oplus a_9 \oplus a_{13} \oplus a_{14} \oplus a_{15} \oplus a_{16} \oplus a_{20}
 \end{aligned}$$

Scan chain 1

$$\begin{aligned}
 & a_1 \oplus a_2 \oplus a_3 \oplus a_6 \\
 & a_0 \oplus a_1 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_8 \\
 & a_2 \oplus a_3 \oplus a_5 \oplus a_6 \oplus a_7 \oplus a_{10} \\
 & a_0 \oplus a_1 \oplus a_4 \oplus a_5 \oplus a_7 \oplus a_8 \oplus a_9 \oplus a_{12} \\
 & a_2 \oplus a_3 \oplus a_6 \oplus a_7 \oplus a_9 \oplus a_{10} \oplus a_{11} \oplus a_{14} \\
 & a_0 \oplus a_5 \oplus a_7 \oplus a_{12} \oplus a_{14} \oplus a_{15} \oplus a_{18} \\
 & a_0 \oplus a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus a_8 \oplus a_9 \oplus a_{12} \oplus a_{13} \oplus a_{15} \oplus a_{16} \oplus a_{17} \oplus a_{20}
 \end{aligned}$$

Scan chain 3

$$\begin{aligned}
 & a_0 \oplus a_1 \oplus a_5 \oplus a_7 \\
 & a_2 \oplus a_3 \oplus a_7 \oplus a_9 \\
 & \underline{a_4 \oplus a_5 \oplus a_9 \oplus a_{11}} \\
 & a_6 \oplus a_7 \oplus a_{11} \oplus a_{13} \\
 & a_1 \oplus a_8 \oplus a_9 \oplus a_{13} \oplus a_{15} \\
 & \underline{a_0 \oplus a_3 \oplus a_{10} \oplus a_{11} \oplus a_{15} \oplus a_{17}} \\
 & a_0 \oplus a_2 \oplus a_5 \oplus a_{12} \oplus a_{13} \oplus a_{17} \oplus a_{19} \\
 & a_2 \oplus a_4 \oplus a_7 \oplus a_{14} \oplus a_{15} \oplus a_{19} \oplus a_{21}
 \end{aligned}$$

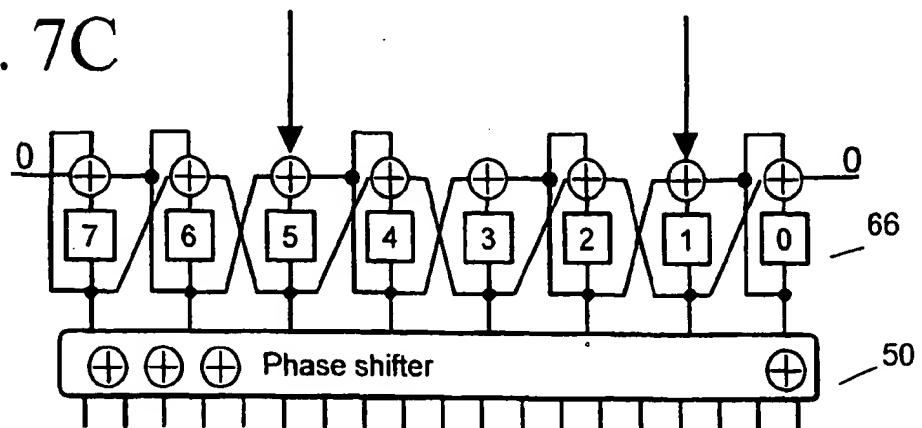
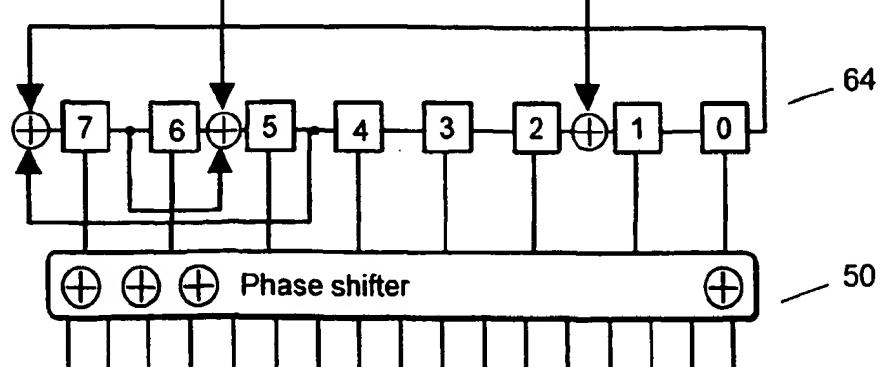
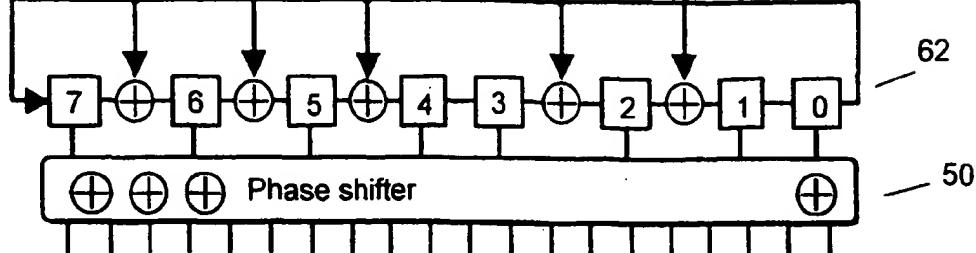
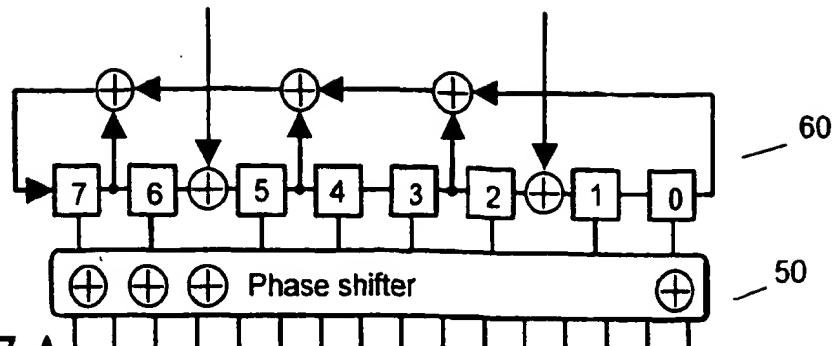
Scan chain 5

$$\begin{aligned}
 & a_0 \oplus a_5 \oplus a_6 \\
 & a_0 \oplus a_1 \oplus a_2 \oplus a_7 \oplus a_8 \\
 & a_2 \oplus a_1 \oplus a_4 \oplus a_9 \oplus a_{10} \\
 & \underline{a_0 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_{11} \oplus a_{12}} \\
 & a_0 \oplus a_1 \oplus a_6 \oplus a_7 \oplus a_8 \oplus a_{11} \oplus a_{13} \\
 & a_0 \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_8 \oplus a_9 \oplus a_{10} \oplus a_{15} \oplus a_{16} \\
 & a_2 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_{10} \oplus a_{11} \oplus a_{12} \oplus a_{17} \oplus a_{18} \\
 & a_1 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_7 \oplus a_8 \oplus a_{12} \oplus a_{13} \oplus a_{14} \oplus a_{19} \oplus a_{20}
 \end{aligned}$$

Scan chain 7

$$\begin{aligned}
 & a_2 \oplus a_3 \oplus a_5 \\
 & a_0 \oplus a_4 \oplus a_5 \oplus a_7 \\
 & a_0 \oplus a_2 \oplus a_6 \oplus a_7 \oplus a_9 \\
 & a_1 \oplus a_2 \oplus a_4 \oplus a_8 \oplus a_9 \oplus a_{11} \\
 & a_0 \oplus a_3 \oplus a_4 \oplus a_6 \oplus a_{10} \oplus a_{11} \oplus a_{13} \\
 & a_1 \oplus a_2 \oplus a_5 \oplus a_6 \oplus a_8 \oplus a_{12} \oplus a_{13} \oplus a_{15} \\
 & a_1 \oplus a_3 \oplus a_4 \oplus a_7 \oplus a_8 \oplus a_{10} \oplus a_{14} \oplus a_{15} \oplus a_{17} \\
 & a_0 \oplus a_3 \oplus a_5 \oplus a_6 \oplus a_9 \oplus a_{10} \oplus a_{12} \oplus a_{17} \oplus a_{19}
 \end{aligned}$$

FIG. 6



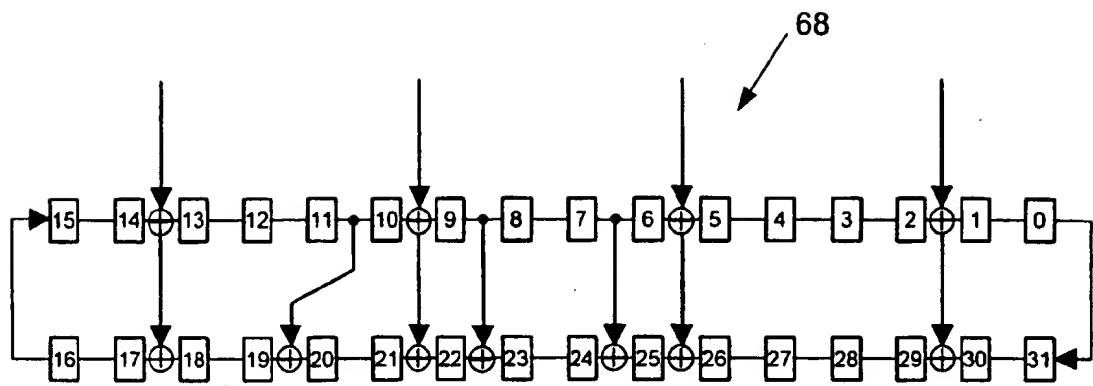


FIG. 8

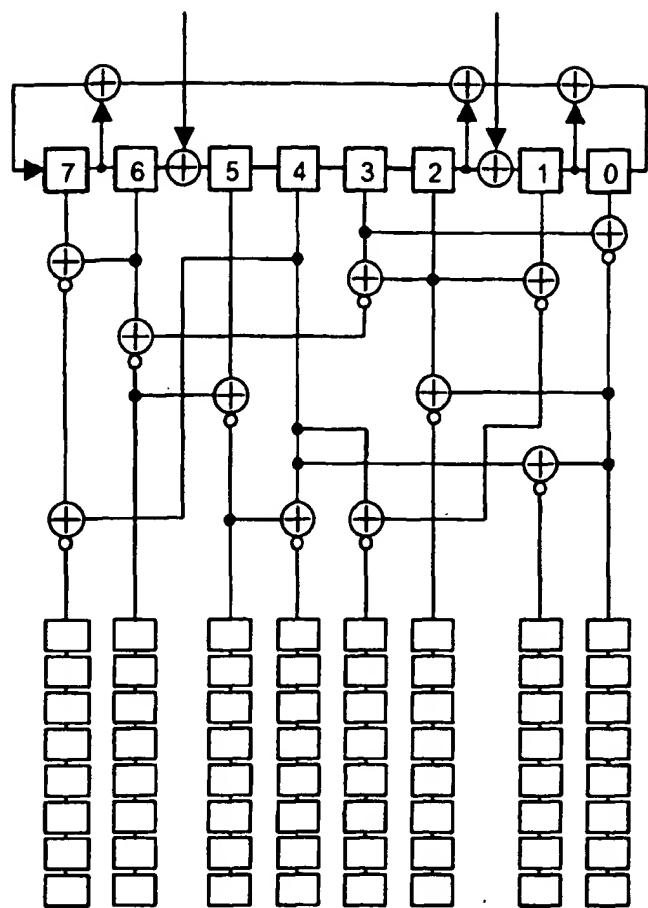


FIG.9

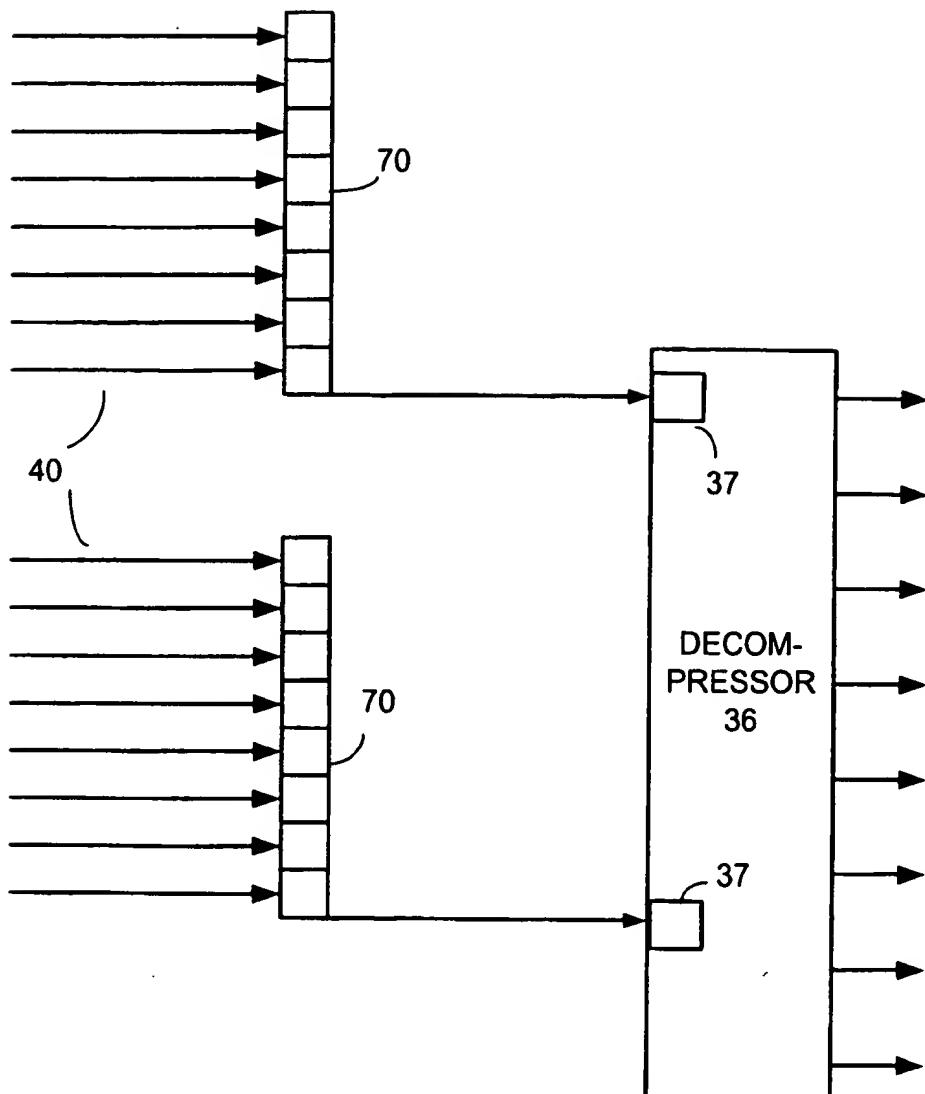


FIG. 10

**REFERENCES CITED IN THE DESCRIPTION**

*This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.*

**Patent documents cited in the description**

- US 5694402 A [0006]

**Non-patent literature cited in the description**

- N. ZACHARIA et al. Decompression of Test Data Using Variable-Length Seed LFSRs. *Proc. VLSI Test Symposium*, 1995, 426-433 [0006]
- J. RAJSKI et al. Test Data Decompression for Multiple Scan Designs with Boundary Scan. *IEEE Trans. on Computers*, 1998, vol. C-47, 1188-1200 [0006]
- L.F.C. LEW YAN VOON. BIST Linear Generator Based on Complemented Outputs. *Proceedings of the 10th IEEE VLSI Test Symposium*, 1992, 137-142 [0006]